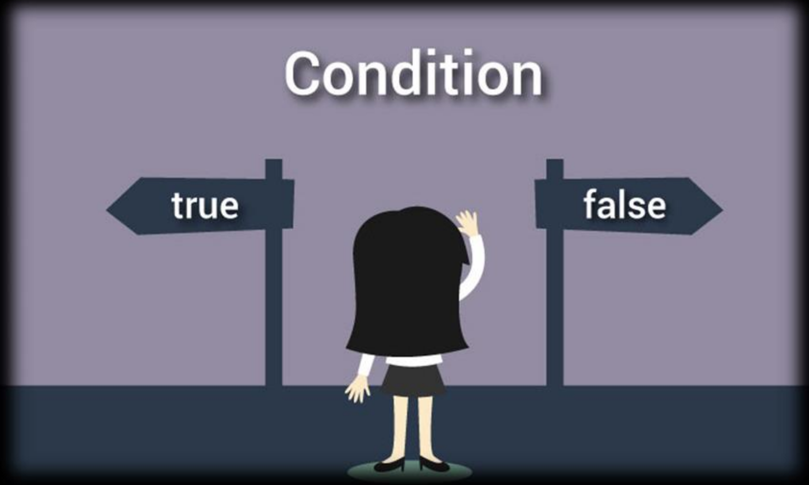
Прости проверки

Логически изрази и проверки
Условна конструкция if-else

Основи на програмирането





СофтУни
трейнърски екип
Софтуерен университет
http://softuni.bg

Have a Question?

SoftUni Foundation

sli.do

#PythonBasics

Съдържание

1. Логически изрази и проверки
   - Оператори за сравнение: `<`, `>`, `==`, `!=`, …
2. Конструкции `if` и `if-else`
3. Серия от проверки – `elif`
4. Дебъгване





Логически изрази и проверки

Оператори за сравнение

Оператори за сравнение

Оператор	Означение	Работи за
Равенство	==	числа, дати, други сравними типове
Различно	!=	
По-голямо	>	
По-голямо или равно	>=	
По-малко	<	
По-малко или равно	<=	

Сравняване на стойности (1)

SoftUni Foundation

- Чрез логически оператори (за числа)

```
a = 5
b = 10
print(a < b)       // true
print(a > 0)       // true
println(a > 100)   // false
print(a < a)       // false
print(a <= 5)      // true
print(b == 2 * a) // true
```

Сравняване на стойности (2)

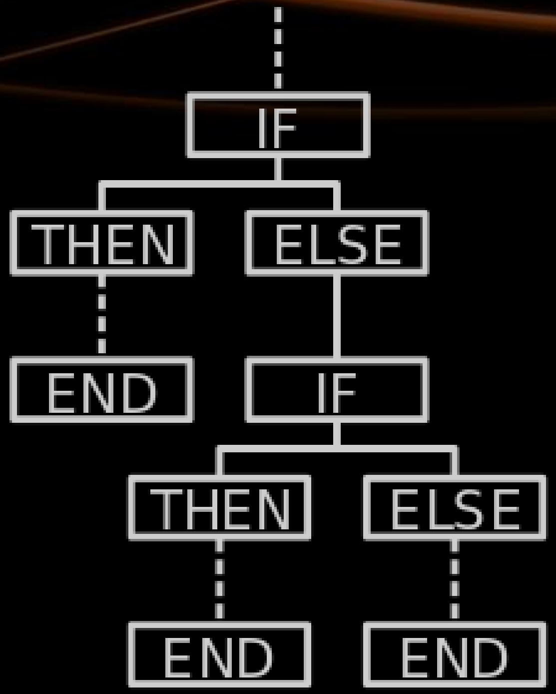
SoftUni Foundation

- Сравняване на текст чрез `==`

```
a = 'Example'
b = a
print(a == b) // true
```

```
a = input()
b = input()
print(a == b); // true
```

Въвеждане на еднаква стойност



Прости проверки

Условни конструкции

Прости проверки

- Често проверяваме условия и извършваме действия според резултата

Условие (булев израз)

Код за изпълнение при вярност на условието

```
if 1 + 2 > 0:
    print('3 > 0')
```

- Резултатът е true или false

Отлична оценка - условие

- Напишете програма, която:
  - Чете оценка (число), въведена от потребителя
  - Проверява дали е отлична
  - Извежда "Excellent", ако оценката е по-голяма или равна на 5,50
- Пример:

Вход		Изход
`4`	→	`няма изход`
`5.50`	→	`Excellent!`

Отлична оценка - решение

SoftUni Foundation

```
grade = float(input())
if grade >= 5.50:
    print('Excellent!')
```

Тестване на решението: https://judge.softuni.bg/Contests/Practice/Index/152#0

Прости проверки – if-else

- При невярност (false) на условието, можем да изпълним други действия – чрез else конструкция

```
if 1 + 2 < 0:
    print('3 < 0')
else:
    print('3 > 0')
```

Код за изпъление при невярност на условието

Блок от код

- Табулациите въвеждат блок от код (група команди)

```
color = 'red'
if color == 'red':
    print('Red')
else:
    print('Yellow')
    println('bye')
```

Извежда се "Red"

Блок от код (2)

- Без тях се изпълнява само първия ред код

```
color = 'red'
if color == 'red':
    print('Red')
else:
    print('Yellow')
print('bye')
```

Извежда се "Red
bye"

Четно или нечетно число – условие

- Напишете програма, която:
  - Проверява дали едно число е **четно** или **нечетно**
  - Ако е четно извежда "**even**"
  - Ако е нечетно извежда "**odd**"
- Пример:

Вход		Изход
4	→	even
7	→	odd

Четно или нечетно – решение

```python
num = int(input())
if num % 2 == 0:
    print('even')
else:
    print('odd')
```

Тестване на решението: https://judge.softuni.bg/Contests/Practice/Index/152#2

По-голямото число – условие

SoftUni Foundation

- Напишете програма, която:
  - Чете две цели числа
  - Извежда "Greater number: "
  - Долепя по-голямото от тях
- Пример

Вход		Изход
5 8	→	8
7 3	→	7

По-голямото число – решение

SoftUni Foundation

```
num1 = int(input())
num2 = int(input())
if num1 > num2:
  print("Greater number: " + str(num1))
else:
  print("Greater number: " + str(num2))
```

Тестване на решението: https://judge.softuni.bg/Contests/Practice/Index/152#3

SoftUni Foundation

```
42
even

Process finished with exit code 0
```

```
5.5
Excellent

Process finished with exit code 0
```

Прости if конструкции

Работа на живо в клас (лаб)



Серии от проверки

По-сложни условни конструкции

Серии от проверки

- Конструкцията **if/elif/else…** може да е в серия

```
if 3 + 1 = 5:
    print('4 = 5')
elif 3 + 2 = 5:
    print('5 = 5')
elif 3 + 3 = 5:
    print('6 = 5')
else:
    print('none')
```

- При истинност на едно условие, не се продължава към проверяване на следващите

Серия от проверки - пример

SoftUni Foundation

```
a = 7
if a > 4:
    print('Bigger than 4')
elif a > 5:
    print('Bigger than 5')
else:
    print('Equal to 7')
```

Извежда се само "Bigger than 4"

Число от едно до 10 с текст - условие

- Напишете програма, която:
  - Чете цяло число, въведено от потребителя
  - Проверява неговата стойност [1,9]
  - Ако числото е по-голямо от 9 извежда "number too big"
  - Извежда стойността с текст
- Пример:

```
7
```

→

```
seven
```

```
10
```

→

```
number too big
```

Число от едно до 10 с текст - решение

SoftUni Foundation

```
num = int(input())
if num == 1:
    print('one')
elif num == 2:
    print('two')
elif num == 3:
    print('three') // TODO: add more checks
else:
    print('number too big')
```

Тестване на решението: https://judge.softuni.bg/Contests/Practice/Index/152#4

Бонус точки – условие

- Напишете програма, която:
  - Чете цяло число – брой точки
  - Изчислява бонус точки и общия брой точки след прилагане на бонусите
  - Принтира сумата
  - Ако числото е:
    - до 100 включително, бонус точките са 5
    - по-голямо от 100, бонус точките са 20%
    - по-голямо от 1000, бонус точките са 10%
  - Допълнителни бонус точки:
    - За четно число → 1 т.
    - За число, което завършва на 5 → 2 т.

Вход	Изход
20	6 26
175	37 212
2703	270.3 2973.3

Бонус точки – решение

SoftUni Foundation

```
num = int(input('Enter score: '))
bonusScore = 0.0

if num > 1000:
  bonusScore = num * 0.10
else: // TODO: write more logic here …

if num % 10 == 5:
  bonusScore += 2
else: // TODO: write more logic here …

print('Bonus score: '+ str(bonusScore))
print('Total score: ' + str(num + bonusScore))
```

Тестване на решението: https://judge.softuni.bg/Contests/Practice/Index/152#5

Сумиране на секунди – условие

SoftUni Foundation

- Трима спортни състезатели финишират за някакъв брой секунди (между 1 и 50). Да се пресметне сумарното им време във формат "**минути:секунди**". Секундите да се изведат с водеща нула (2 → "02", 7 → "07", 35 → "35").
- Примери:

Вход	Изход
35 45 44	2:04
22 7 34	1:03
50 50 49	2:29
14 12 10	0:36

Сумиране на секунди – решение

```
sec1 = int(input())
// TODO: Read also sec2 and sec3 …
secs = sec1 + sec2 + sec3
mins = 0
if secs > 59:     // TODO: Repeat this 2 times …
    mins++;
    secs = secs - 60
if secs < 10:
  print(str(mins) + ":" + '0' + str(secs))
else:
  print(str(mins) + ':' + str(secs))
```

Тестване на решението: https://judge.softuni.bg/Contests/Practice/Index/152#6

Конвертор за мерни единици – условие

- Да се напише програма, която преобразува разстояние между посочените в таблицата мерни единици:
  - Вход: число, входна мерна единица, изходна мерна единица
  - Примерен вход и изход:

```
12
km
ft
```

```
39370.0788 ft
```

входна единица	изходна единица
1 meter (m)	1000 millimeters (mm)
1 meter (m)	100 centimeters (cm)
1 meter (m)	0.000621371192 miles (mi)
1 meter (m)	39.3700787 inches (in)
1 meter (m)	0.001 kilometers (km)
1 meter (m)	3.2808399 feet (ft)
1 meter (m)	1.0936133 yards (yd)

Конвертор за мерни единици – решение

```
size = float(input())
sourceMetric = input()
destMetric = input()
if sourceMetric == 'km':
    size = size / 0.001
// TODO: mm, cm, ft, yd, ...
if destMetric == 'ft':
    size = size * 3.2808399
// TODO: mm, cm, ft, yd, ...
print(str(size) + ' ' + str(destMetric))
```

Тестване на решението: https://judge.softuni.bg/Contests/Practice/Index/152#7

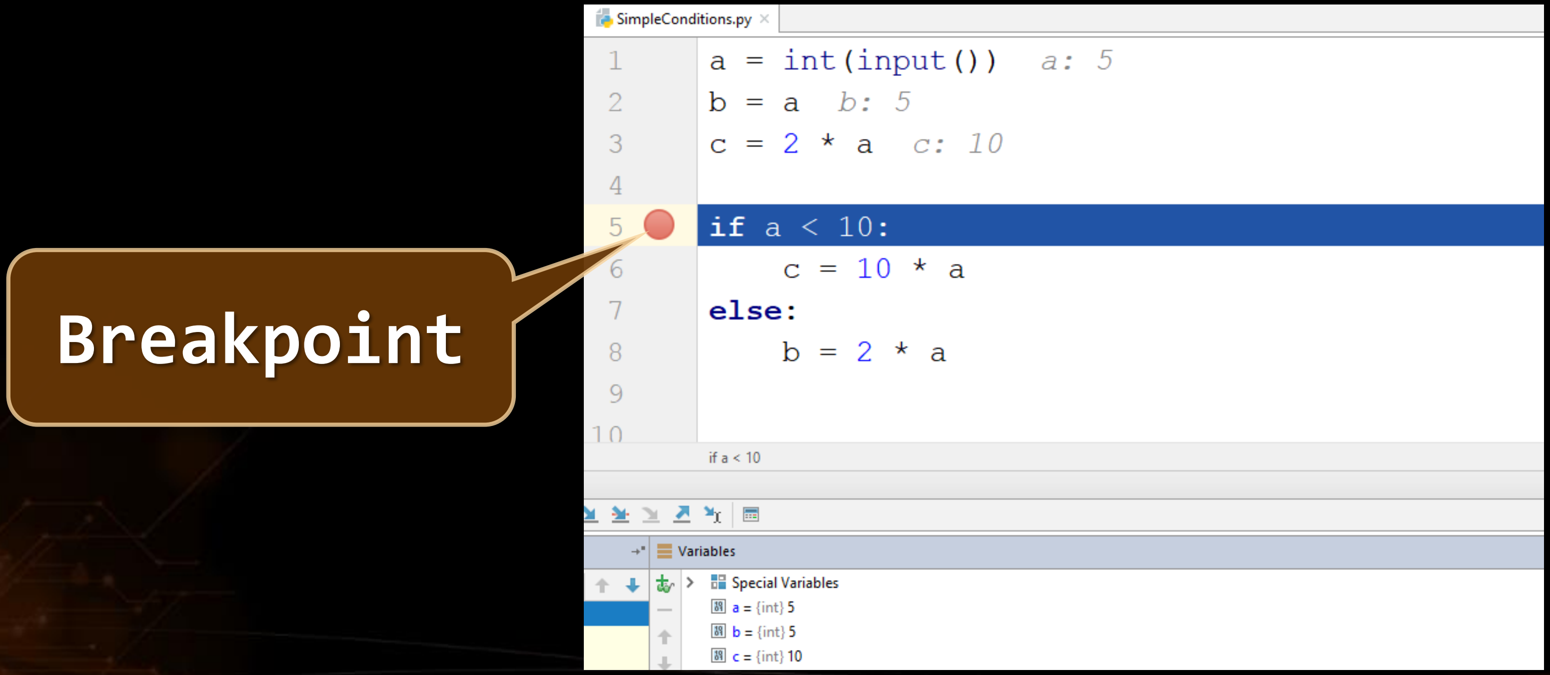


Дебъгване

Прости операции с дебъгер

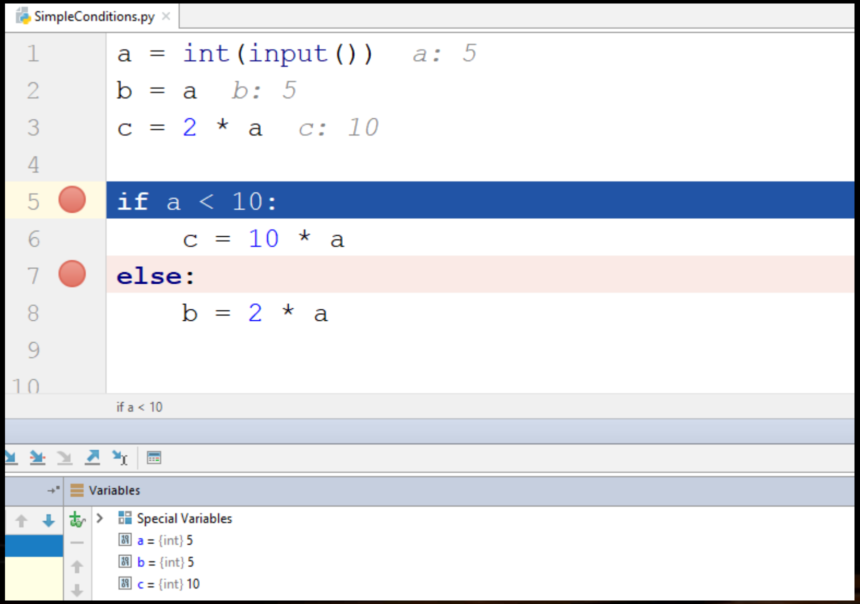
Дебъгване

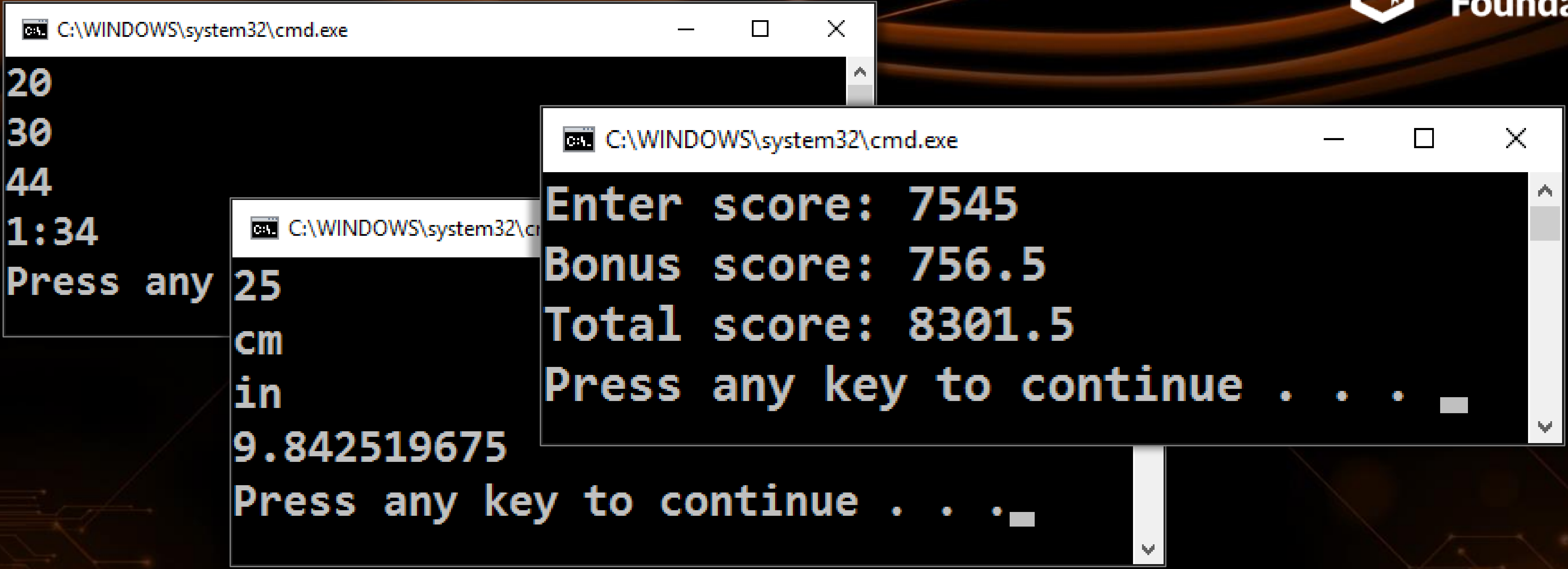
- Процес на проследяване на изпълнението на програмата, което ни позволява да проследим процеса на изпълнение
  - Това ни позволява да откриваме грешки (бъгове)



Дебъгване в PyCharm

- Натискане на [Shift + F9] ще стартира програмата в debug режим
- Можем да преминем към следващата стъпка с [F8]
- Можем да създаваме [Ctrl + F8] стопери – breakpoints
  - До тях можем директно да стигнем изпозлвайки [F9]





SoftUni Foundation

Задачи с прости проверки

Работа на живо в клас (лаб)

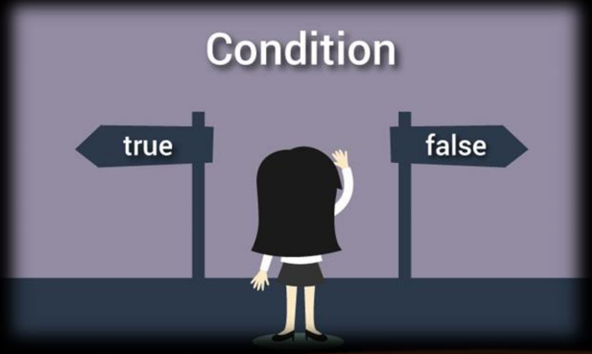
Какво научихме днес?

- Конструкции за проверка на условие **if** и **if-else**:

```
if условие:
  група команди
elif условие2:
  група команди
else:
  група команди
```

```
if условие
  единична_команда
elif условие2:
  единична_команда
elif условие3:
  единична_команда
else:
 единична_команда
```





Прости проверки



Въпроси?

https://softuni.bg/courses/programming-basics/

Лиценз

SoftUni Foundation

- Настоящият курс (слайдове, примери, видео, задачи и др.) се разпространяват под свободен лиценз "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International"

- Благодарности: настоящият материал може да съдържа части от следните източници
  - Книга "Основи на програмирането със Java" от Светлин Наков и колектив с лиценз CC-BY-SA

Безплатни обучения в СофтУни

- Фондация "Софтуерен университет" – softuni.org
- Софтуерен университет – качествено образование, професия и работа за софтуерни инженери
  - softuni.bg
- СофтУни @ Facebook
  - facebook.com/SoftwareUniversity
- СофтУни форуми – forum.softuni.bg

